LeaderBoard & Yesterday's Solution(/faces/candidate/leaderboarddailychallenge.xhtml?RT=DAILYCHALLENGE)

Daily Challenge **Happy Coding from necse** **SkillRack**

Emptying Water from Buckets

There are **N buckets** arranged in a row. Each bucket has a certain amount of water. The maximum capacity and the amount of water in each bucket are passed as the input. A boy performs **N-1 operations** based on the following conditions.
- In the first operation, he empties the 1st bucket into the 2nd bucket (i.e., pouring water from the 1st bucket into the 2nd bucket).
- In the second operation, he empties the 2nd bucket into the 3rd bucket.
- Similarly, he performs the remaining operations.
- During the emptying operation, if the next bucket is full and some water is remaining in the current bucket, the water will be kept in the same bucket.
After N-1 operations, the program must print the amount of water in the last bucket and the total amount of water remaining in the first N-1 buckets.

Boundary Condition(s):
1 <= N <= 100
For each bucket, 0 <= Amount of water <= Maximum capacity <= 10^5

Input Format:
The first line contains N.
The second line contains N integers separated by a space representing the maximum capacities of the N buckets.
The third line contains N integers separated by a space representing the amount of water in the N buckets.

Output Format:
The first line contains two integers separated by a space representing the amount of water in the last bucket and the total amount of water remaining in the first N-1 buckets.

Example Input/Output 1:
Input:
3
3 4 5
1 3 4

Output:
5 3

Explanation:
Initially, the amount of water in the three buckets are **[1 3 4]**.
1st operation: 1st bucket -> 2nd bucket
[0, 4, 4]
2nd operation: 2nd bucket -> 3rd bucket
[0, 3, 5]
The amount of water in the last bucket is **5**.
The total amount of water in the first two buckets is **3** (0 + 3).

Example Input/Output 2:
Input:
3
3 2 3
0 0 0

Output:
0 0

Example Input/Output 3:
Input:
4
10 20 30 40
2 2 2 2

Output:
8 0

Max Execution Time Limit: 50 millisecs

Ambiance

Java (12.0)

```
import java.util.*;public class Hello {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] cap=new int[n];
        int[] con=new int[n];
        for(int i=0;i<n;i++){
          cap[i]=sc.nextInt();
        }
        for(int i=0;i<n;i++){
          con[i]=sc.nextInt();
        }
    //System.out.print(Arrays.toString(cap));
    //System.out.print(Arrays.toString(con));
        for(int i=0;i<n-1;i++){
            int emp=cap[i+1]-con[i+1];
            if(emp>=con[i]){
                con[i+1]+=con[i];
                con[i]=0;
            }
            else if(emp<con[i]){
                con[i+1]+=emp;
                con[i]=con[i]-emp;
            }


        }
int sum=0;
        for(int i=0;i<n-1;i++){
            sum+=con[i];
        }

        System.out.print(con[n-1]+" "+sum);









    }


}
```

1912080@nec

Code did not pass the execution

Hello.java:29: error: cannot find symbol

```
        sum+=con[i];
        ^
```

symbol: variable sum

location: class Hello

```
Hello.java:32: error: cannot find symbol
        System.out.print(con[n-1]+" "+sum);
                                     ^
  symbol:   variable sum
  location: class Hello
2 errors
```

Save Run